

Nodes in GMPLS Lighwave Agile Switching Simulator (GLASS)

Version: Draft 1.0

TABLE OF CONTENTS

1	INTRODUCTION	1
2	PROTOCOL GRAPH	2
3	HOST AND ROUTER	3
3.1	HOST.....	3
3.2	ROUTER	4
4	THE NODES IN GLASS.....	4
4.1	EXTROUTER	4
4.2	LABEL SWITCHING ROUTER (LSR).....	5
4.3	OXCEDEROUTER.....	6
4.4	OXC	6
5	DML SCHEMAS.....	7
6	ACRONYMS	8
7	REFERENCES	8

TABLE OF FIGURES

Figure 1: Component hierarchy	2
Figure 2: Host	3
Figure 3: ExtRouter.....	4
Figure 4: Label Switching Router.....	5
Figure 5: OxcEdgeRouter.....	6
Figure 6: Optical Cross-Connect (OXC).....	6

1 INTRODUCTION

The GLASS Simulator framework extends the Scalable Simulator Framework (SSF) and its extension SSFNet [1]. The nodes in a topology take an important place in the behavior of a system. SSFNet provides some typical nodes (host and router), and the GLASS framework adds extended routers and optical switches to handle optical networks.

This document will present, one a first part, the common features of all the nodes then it will show the nodes provided by SSFNet, and on a last part the nodes available in GLASS.

2 PROTOCOL GRAPH

This section shows the hierarchy of the nodes and explains the basics of the nodes in SSFNet.

As shown in the following figure shows the class hierarchy, **SSF.OS.ProtocolGraph** is the super class of all the other elements. The **ProtocolGraph** must be seen as a container of **ProtocolSession** [2]. Each of these **ProtocolSession** is the representation of a protocol (for example IP, TCP...) and also the network interface cards (**NIC**, **_NIC**, **ONIC**).

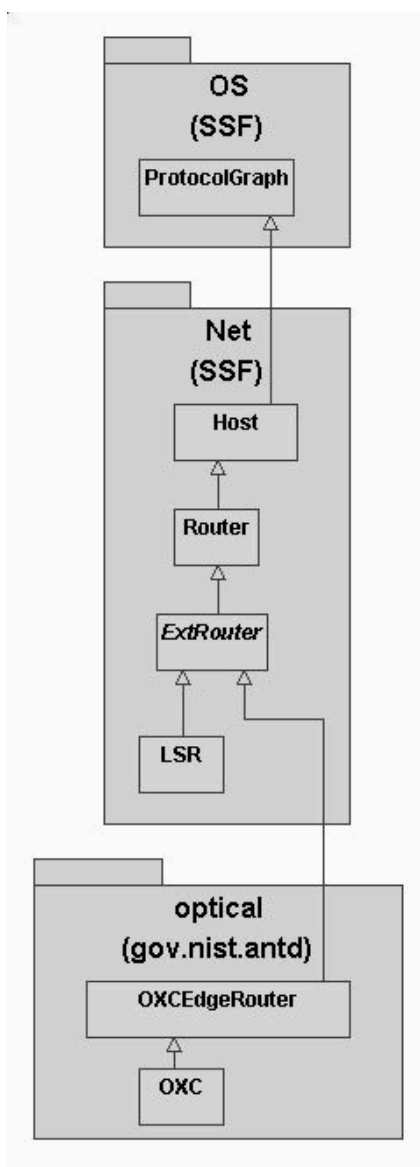


Figure 1: Component hierarchy

The **ProtocolGraph** is in charge of configuring all the **ProtocolSessions** and also provides mechanism to retrieve the protocols during the simulation.

3 HOST AND ROUTER

The Host and Routers are the nodes provided by SSFNet.

3.1 HOST

The following figure shows the UML diagram of the Host class.

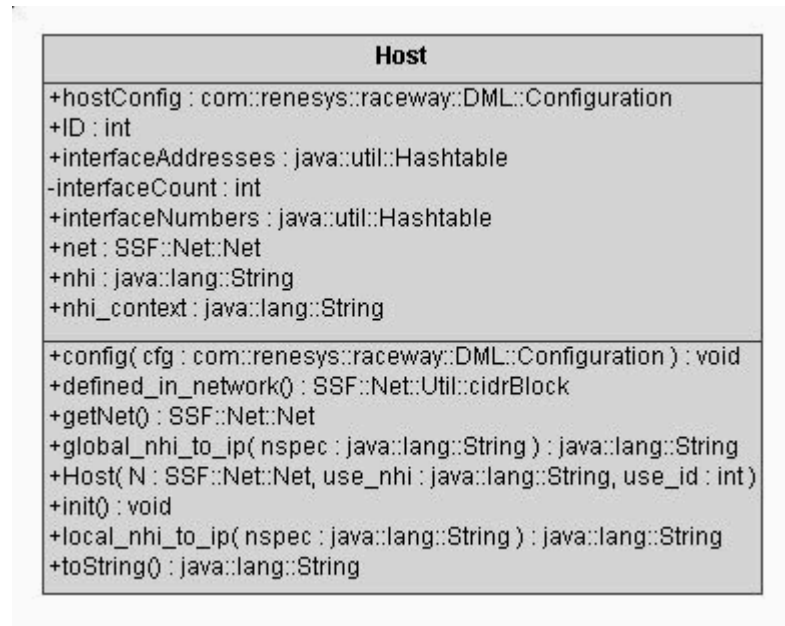


Figure 2: Host

Class **Host** is actually derived from **SSF.OS.ProtocolGraph**, which means that it is fully configurable and can support any network protocol graph. Minimally, this must include IP (which provides routing) and at least one **NIC**.

The **Host** contains an ID that must be unique in the net. It also provides an easy access to the network interface cards.

The initialization is automatically realized when starting a simulation through the call to the method *public void init()*.

Note: in the GLASS simulator we modified the default class for the interface cards. Instead of creating **SSF.Net.NIC**, the framework creates **SSF.Net._NIC**. The class **_NIC** extends **NIC** so that it

provides a compatibility with any examples but also include enhance features (including failure capability, failure notification, input traffic monitoring).

3.2 ROUTER

A Router is a special case of a Host with multiple interface cards, and possibly a specialized protocol graph. There is no other distinction between the Router and Host. Practically, it is also possible to have multiple interfaces connected to one Host.

4 THE NODES IN GLASS

This section shows the different nodes that have been created in the GLASS framework and that can be used when creating optical networks.

4.1 EXTROUTER

The **SSF.Net.ExtRouter** is an abstract class that is used as a base for all the nodes that needs to handle optical network interface cards (**ONIC**). Of course the regular **NICs** are also included in this node.

Below is the UML diagram of the ExtRouter:

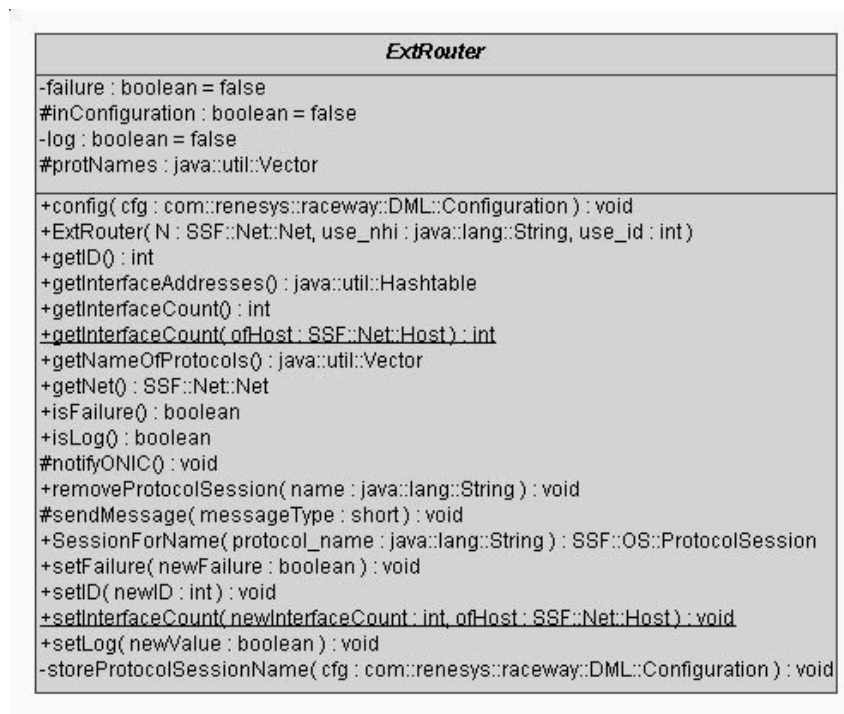


Figure 3: ExtRouter

The **ExtRouter** has been created to provide the basic handle of **ONICs**. It extends the Router so it is fully configurable.

A failure attribute can be set to specify that the node is failed or recovered. In the current framework, triggering the failure attribute will cause all the optical interfaces on the peer side of the links to check their status.

The **ExtRouter** is the default node used by the algorithm provided in the package `gov.nist.antd.merlin.algorithm`. This means that any of the node that subclass the `ExtRouter` will be included for the computation of routes and wavelength assignment.

4.2 LABEL SWITCHING ROUTER (LSR)

The **SSF.Net.LSR** is a specialized node that handles GMPLS layer [3]. It extends the **ExtRouter** in order to handle optical interface cards.



Figure 4: Label Switching Router

This node does not have any specific configuration and can also be used as a regular Router if needed.

4.3 OXCEdgeRouter

The **gov.nist.antd.optical.OXCEdgeRouter** is the first node located in the optical package. It has been created to provide switching capabilities of optical messages between optical links [5].

It can be used at the edge between non-optical network and the optical core.

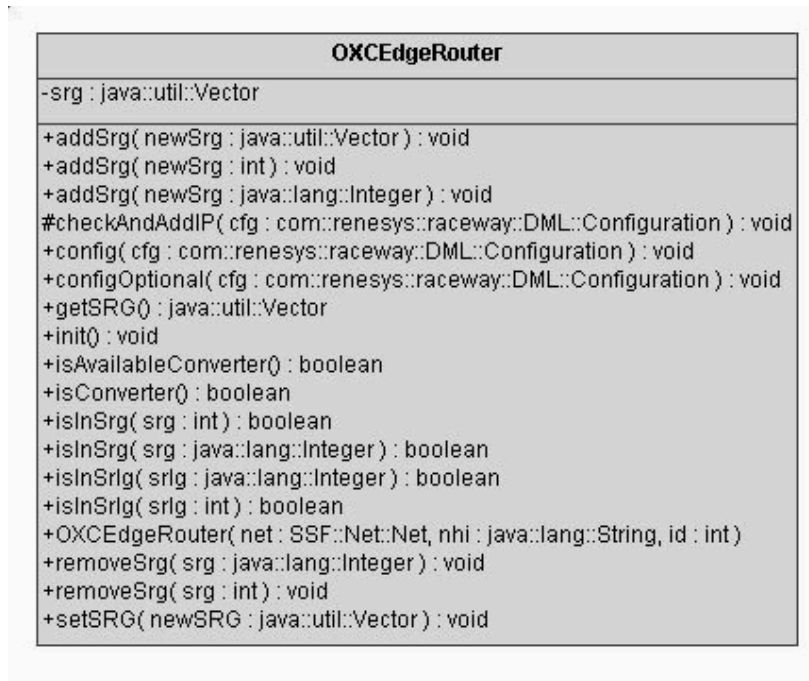


Figure 5: OxcEdgeRouter

The OXCEdgeRouter includes Shared Risk Group (SRG) that can be used by algorithms to compute disjoint routes. The SRG is used in addition of the Shared Risk Ling Group (SRLG) of the links [4].

4.4 OXC

The **gov.nist.antd.optical.OXC** is a representation of an optical cross-connect. It is a specialized node that is used in the optical core network. It only provides switching capabilities and cannot accept regular interfaces cards.

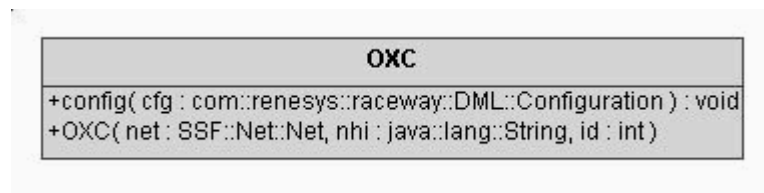


Figure 6: Optical Cross-Connect (OXC)

5 DML SCHEMAS

This section provides the DML schema of the nodes. The schema for Host, Router and Graph are a copy of the documentation provided by SSF [5].

<pre>host [id %I idrange [from %I1! to %I1!] alignment %S]</pre>	<p>Net.host specifies a machine with a protocol graph installed, and zero or more configured network interfaces. (Net.router has identical attributes.) Must contain either a single integer ID (id) (or a range of contiguous, ascending integer IDs (idrange) but not both). Relative to the immediately enclosing Net context the id and idrange values must be unique for each host and router.</p>
<pre>graph [description %S ProtocolSession [name %S1! use %S1!]]</pre>	<p>Net.host.graph is an internal attribute specifying a list of protocols to be configured and the names of SSF.OS classes to use. graph.description is an optional protocol graph description string, ProtocolSession.name is a symbolic tag, like "tcp" or "ip", and ProtocolSession.use is the name of a class extending SSF.OS.ProtocolSession that will be installed. Use one ProtocolSession for each protocol.</p>
<pre>router [_extends .schemas.Net.host]</pre>	<p>Syntactically, a router is synonymous with host, but it will usually have distinct protocols in its protocol graph, and optional configuration attributes required by OSPF or other routing protocols. The different roles of routers and hosts in a simulation warrant using different names.</p>
<pre>LSR [_extend .schemas.Net.router failure %S LogFile %S]</pre>	<p>The LSR is an instantiation of the ExtRouter, and can have optical network interface cards (ONICs). The optional attribute failure comes from ExtRouter (default value: false). The optional attribute LogFile is used by the GMPLS implementation to create output file. To activate it, the value must be "on". If not précised or not equal to this value, no output file (for traffic monitoring) will be created.</p>
<pre>OXCEdgeRouter [_extend .schemas.Net.router]</pre>	<p>The OXCEdgeRouter is an instantiation of the ExtRouter, and can have optical</p>

<pre>failure %S srg %S]</pre>	<p>network interface cards (ONICs). The optional attribute failure comes from ExtRouter (default value: false). Optional attribute srg : The Shared Risk Group. This attribute is an integer list that represents all SRG this node share. The integers are separated by comma (default value: null).</p>
<pre>OXC [_extend .schemas.Net.OXCEdgeRouter]</pre>	<p>The OXC is a restricted version of the OXCEdgeRouter because it cannot contain regular interfaces (NIC) but only Optical network interface cards (ONIC).</p>

6 ACRONYMS

GLASS	GMPLS Lightwave Agile Switching Simulator
SSF	Scalable Simulation Framework
IP	Internet Protocol
LSR	Label Switching Router
OXC	Optical Cross-Connect
SRG	Shared Risk Group
NIC	Network Interface Card
ONIC	Optical Network interface Card

7 REFERENCES

- [1] SSFNet
By Renesys Corporation
URL: <http://www.ssfnet.com/>

- [2] ProtocolSessions in the GMPLS Lightwave Agile Switching Simulator (GLASS)
By NIST/ANTD.
URL: <http://www.antd.nist.gov/glass>

- [3] GMPLS implementation in GMPLS Lightwave Agile Switching Simulator (GLASS)
By NIST/ANTD (not released yet).
URL: <http://www.antd.nist.gov/glass>

- [4] Links in GMPLS Lightwave Agile Switching Simulator (GLASS)
By NIST/ANTD.
URL: <http://www.antd.nist.gov/glass>

- [5] Standard SSFNet DML attributes
By Renesys Corporation
URL: <http://www.ssfnet.com/InternetDocs/ssfnetDMLReference.html>